Homework 9

The purpose of this homework is to learn more about multiple regression models and data wrangling. Please fill in the appropriate code and write answers to all questions in the answer sections, then submit a compiled pdf with your answers through gradescope by 11:59pm on Sunday November 17th.

As always, if you need help with any of the homework assignments, please attend the TA office hours which are listed on Canvas and/or ask questions on Piazza. Also, if you have completed the homework, please help others out by answering questions on Piazza.

Part 1: Polynomial regression

In the first set of exercises you will get practice running polynomial regression using the IPEDS faculty salary data.

Part 1.1 (2 points): To start, use dplyr to create a data frame called `IPED_2` that only includes schools that have endowments greater that 0 dollars. Also, add a variable to this data frame called `log_endowment` which is the log10 of each school's endowment.

```
load('IPED_salaries_2016.rda')
IPED_2 <- IPED_salaries %>% filter(endowment > 0) %>% mutate(log_endowment = log10(endowment))
```

Part 1.2 (5 points): Fitting polynomial models

Now use polynomial regression to build models that predict total faculty salaires (salary_tot) from log_endowment. Do the polynomial fit for models up to degree 5, and for every model be sure to include the lower order terms as well; i.e., the model of degree 3 should be $\hat{y} = \hat{\beta_0} + \hat{\beta_1}x + \hat{\beta_2}x^2 + \hat{\beta_3}x^3$. Save all these models in a list called `poly_models`. Then use the summary() function to print the model of degree 5. Report which coefficients appear to be statistically significant from the degree 5 model.

```
poly_models <- list()
for (i in 1:5){
  poly_models[[i]] <- lm(salary_tot ~ poly(log_endowment, degree = i), data = IPED_2)
}
summary(poly_models[[5]])
```

```
##
## Call:
## lm(formula = salary_tot ~ poly(log_endowment, degree = i), data = IPED_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -97488 -12157  -2362   9270 167196
##
## Coefficients:
##                                   Estimate Std. Error t value
## (Intercept)                        64565.3      258.1 250.196
## poly(log_endowment, degree = i)1 1067211.9    20672.8  51.624
## poly(log_endowment, degree = i)2  437640.8    20693.8  21.148
## poly(log_endowment, degree = i)3  180102.7    20706.2   8.698
## poly(log_endowment, degree = i)4  -68788.4    20726.7  -3.319
## poly(log_endowment, degree = i)5  -39140.9    20691.5  -1.892
##                                              Pr(>|t|)
## (Intercept)                      < 0.0000000000000002 ***
## poly(log_endowment, degree = i)1 < 0.0000000000000002 ***
## poly(log_endowment, degree = i)2 < 0.0000000000000002 ***
## poly(log_endowment, degree = i)3 < 0.0000000000000002 ***
## poly(log_endowment, degree = i)4             0.000909 ***
## poly(log_endowment, degree = i)5             0.058584 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20630 on 6386 degrees of freedom
##   (6 observations deleted due to missingness)
## Multiple R-squared:  0.3335, Adjusted R-squared:  0.333
## F-statistic: 639.1 on 5 and 6386 DF,  p-value: < 0.00000000000000022
```

Answer:

All of the degrees seem to be statistically significant except degree 5, which has a p-value of 0.058584.

Part 1.3 (7 points): Plotting polynomial models

Now visualize these fits of these different polynomial models (i.e., the $\hat{y}$ lines) by creating a scatter plot of the faculty salaries as a function of $log_{10}(endowment)$. Then run a for loop to plot a line for each model fit by:

1) predicting the salaries from a model of the current degree
2) plotting the predicted values as a function of the log_10 endowment in a distinct color (creating a vector with color names outside of the for loop will be helpful).

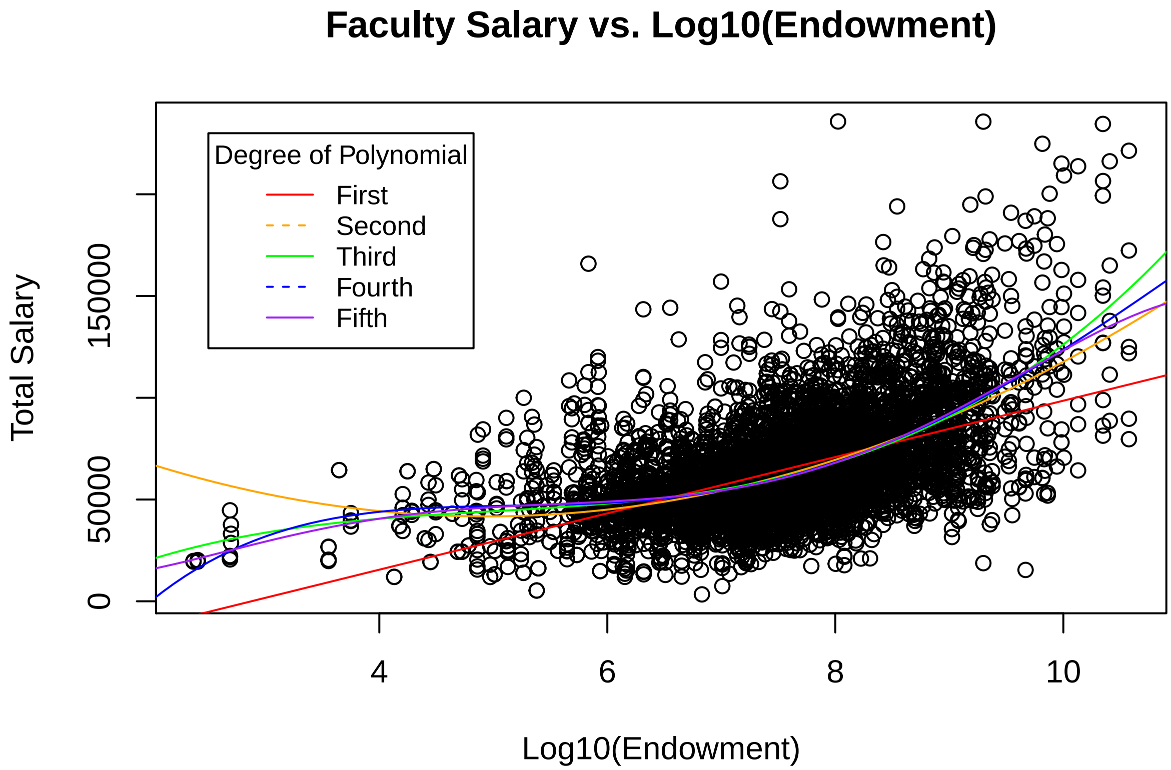
Try to add a legend to the plot showing what the different colored lines correspond to, and report below which model seems to be the best fit.

```r
# create a data frame for making predictions and a vector of colors
predict_df <- data.frame(log_endowment = seq(0, 13, by = .1))
the_cols <- c("red", "orange", "green", "blue", "purple")

# plot the original data
plot(IPED_2$log_endowment, IPED_2$salary_tot, xlab = "Log10(Endowment)",
     ylab = "Total Salary", main = "Faculty Salary vs. Log10(Endowment)")

# plot each of the model fits
for(i in 1:5){
  y_vals_predicted <- predict(poly_models[[i]], newdata = predict_df)
  points(predict_df$log_endowment, y_vals_predicted,
      type = "l", col = the_cols[i])
}

#Added a legend
legend(2.5, 230000, legend=c("First", "Second", "Third",
                             "Fourth", "Fifth"),
       col=c("red", "orange", "green", "blue", "purple"),
       title = "Degree of Polynomial", lty=1:2, cex=0.8)
```



Answer
From the plot above, it seems that the fifth degree polynomial is still the best fit.

Part 1.4 (5 points): Extracting Rˆ2 and adjusted Rˆ2 statistics

Now extract the R^2 and adjusted R^2_{adj} statistics. Which model has the largest the R^2 and the largest adjusted R^2_{adj} statistics? Is this what you would expect.

```
all_r_squared <- c()
all_r_adj <- c()
for (i in 1:5){
  all_r_squared[i] <- summary(poly_models[[i]])$r.squared
  all_r_adj[i] <- summary(poly_models[[i]])$adj.r.squared

}
all_r_squared
```

```
## [1] 0.2774779 0.3240210 0.3319748 0.3331177 0.3334912
```

```
all_r_adj
```

```
## [1] 0.2773648 0.3238094 0.3316611 0.3327000 0.3329693
```

Answer:
The fifth degree polynomial has both the largest R^2 and R^2_{adj} statistics. I expected the R^2 to be the largest, but I thought the addition of another degree would decrease the adjusted value.

Part 1.5 (3 points): Do these models seem reasonable?

Describe overall whether you feel fitting polynomial models here seems like a reasonable thing to do; i.e., pro and cons of using a polynomial model here. There is not necessarily a right answer, just express your thoughts.

Answer
I think fitting a polynomial here does make sense because in reality, all salaries will be within a certain range (above 0 and below some number). This means we shouldn't just use a linear model to predict but should find a model which will account for the flattening of values near the ends of regression. However, there is a disadvantage in that using a polynomial will force the regression into some shape which may not necessarily reflect the nature of salary vs. endowment.

Part 2: Exploring categorical predictors and interactions

Let's now examine how much faculty salaries increase as a function of log endowment size taking into account the rank that different professors have.

Part 2.1 (2 points): Wrangling the data

Start this analysis by creating a data set called `IPED_3` which is modified `IPED_2` in the following way:

1) Only include data from institutions with a CARNEGIE classification of 15 or 31 (these correspond to R1 institutions and liberal arts colleges).

2) Only use the faculty ranks of Lecturer, Assistant, Associate, and Full professors

If you do this right you should `IPED_3` should have 808 rows.

```
IPED_3 <- IPED_2 %>% filter(CARNEGIE %in% c(15,31),
         rank_name %in% c("Lecturer", "Assistant", "Associate", "Full"))
dim(IPED_3)
```

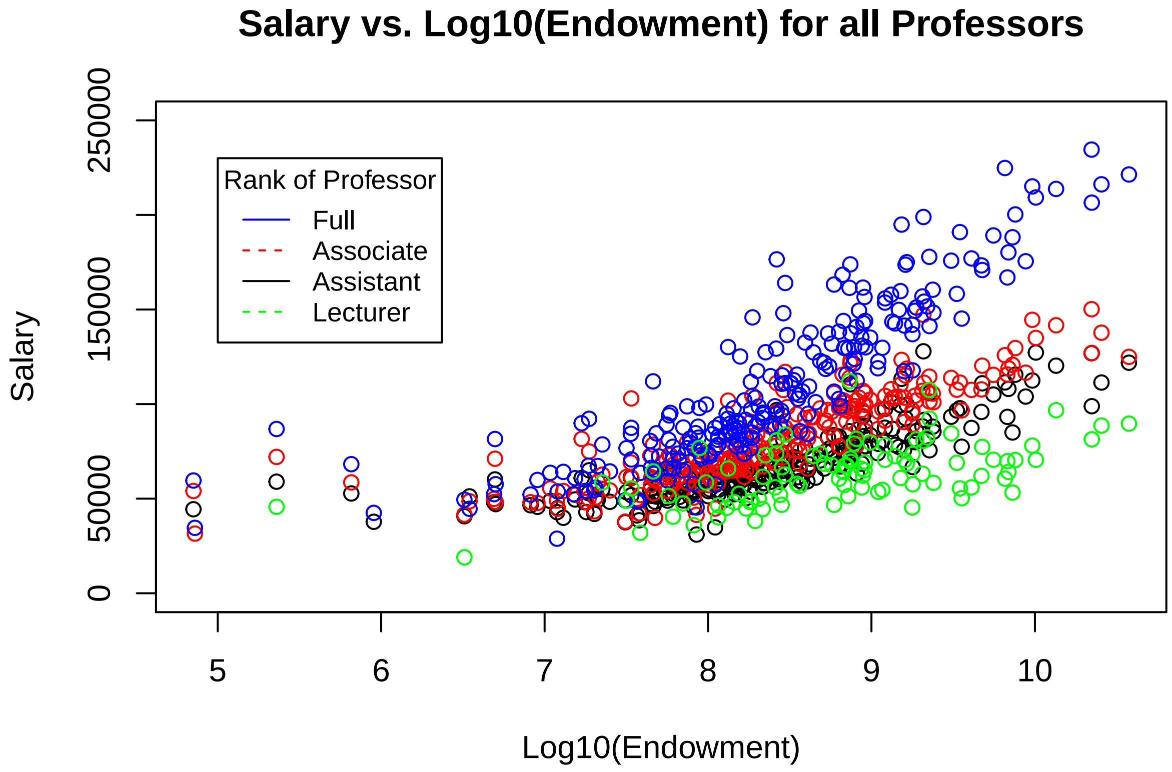
```
## [1] 808  15
```

Part 2.2 (3 points): Visualizing the data

Now create a scatter plot of the data showing the total salary that faculty get paid (salary_tot) as a function of the log endowment size, where each faculty rank is in a different color. In particular, use the following color scheme:

a) Assistant professors are in black
b) Associate professors are in red
c) Full professors are in blue
d) Lecturers are in green

```r
plot(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Assistant"),
     ylim = c(0, 250000), ylab = "Salary", xlab = "Log10(Endowment)",
     main = "Salary vs. Log10(Endowment) for all Professors")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Associate"), col = "red")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Full"), col = "blue")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Lecturer"), col = "green")
legend(5, 230000, legend=c("Full", "Associate", "Assistant",
                           "Lecturer"),
       col=c("blue", "red", "black", "green"),
       title = "Rank of Professor", lty=1:2, cex=0.8)
```



Part 2.3 (7 points): Fitting a linear model to the data

Now fit a linear regression model for total salary as a function of log endowment size, but use a separate y-intercept for each of the 4 faculty ranks (and use the same slope for all ranks).

Use the summary() function to extract information about the model, and then answer the following questions about the model:

1) How much do faculty salaries increase for each order of magnitude increase in endowment size?
2) What is the reference faculty rank that the other ranks are being compared to?
3) What is the difference in faculty salaries for each of the other ranks relative to the reference rank?
4) Do there appear to be statistically significant differences between the y-intercept of reference rank and each of the other ranks?
5) How much of the total sum of squares of faculty salary is log10 endowment and faculty rank accounting for in this model based on the R^2 and adjusted R^2 statistics?

```
(intercept_fit <- lm(salary_tot ~ log_endowment + rank_name, data = IPED_3))
```

```
##
## Call:
## lm(formula = salary_tot ~ log_endowment + rank_name, data = IPED_3)
##
## Coefficients:
##        (Intercept)       log_endowment  rank_nameAssociate
##            -106023               25797              -27824
## rank_nameAssistant   rank_nameLecturer
##             -40934              -57334
```

```
summary(intercept_fit)
```

```
##
## Call:
## lm(formula = salary_tot ~ log_endowment + rank_name, data = IPED_3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -53203 -11161  -2275   7280  77600
##
## Coefficients:
##                     Estimate Std. Error t value             Pr(>|t|)
## (Intercept)        -106022.8     6359.8  -16.67 <0.0000000000000002 ***
## log_endowment        25796.8      748.6   34.46 <0.0000000000000002 ***
## rank_nameAssociate  -27823.8     1685.2  -16.51 <0.0000000000000002 ***
## rank_nameAssistant  -40933.9     1685.2  -24.29 <0.0000000000000002 ***
## rank_nameLecturer   -57334.4     2236.6  -25.64 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 18360 on 803 degrees of freedom
## Multiple R-squared:  0.7053, Adjusted R-squared:  0.7038
## F-statistic: 480.5 on 4 and 803 DF,  p-value: < 0.00000000000000022
```

```r
(the_coefs_salary <- coef(intercept_fit))
```

```
##        (Intercept)      log_endowment rank_nameAssociate
##         -106022.78           25796.82          -27823.77
## rank_nameAssistant  rank_nameLecturer
##          -40933.90          -57334.39
```

Answers

1) The faculty salary increases $25796.82 for each order of magnitude increase in endowment size.

2) The reference faculty is the Full Professor.

3) Associate Professors are paid $27923.77 less, Assistant Professors are paid $40933.90 less, and Lecturers are paid $57334.39 less.

4) The p-values are all very close to 0 (about $2 * 10^{-16}$), so it appears there is a statistically significant difference between the y-intercept of each rank versus the Full Professor.

5) $R^2 = 0.7053$ and adjusted $R^2 = 0.7038$. So the model accounts for 70.38% of the sum of squares (using the adjusted R-squared).

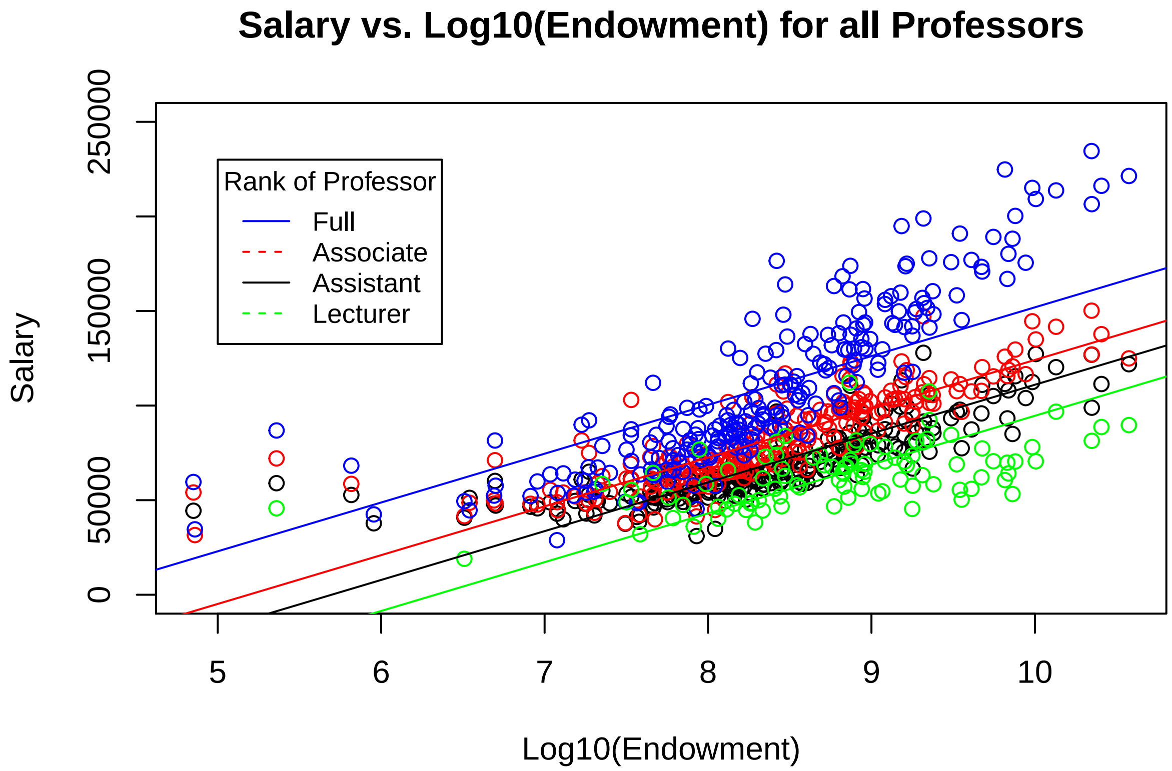
Part 2.4 (5 points): Visualizing the model fit

Now recreate the scatter plot you created in part 2.2 using the same colors. Now, however, also add on the regression lines with different y-intercepts that you fit in part 2.4 (using the appropriate colors to match the colors of the points).

Are there any situations in particular where using the same slope for each rank seem like it is doing a poor job fitting the data?

```r
plot(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Assistant"),
     ylim = c(0, 250000), ylab = "Salary", xlab = "Log10(Endowment)",
     main = "Salary vs. Log10(Endowment) for all Professors")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Associate"), col = "red")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Full"), col = "blue")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Lecturer"), col = "green")

abline(the_coefs_salary[1], the_coefs_salary[2], col = "blue")
abline(the_coefs_salary[1] + the_coefs_salary[3], the_coefs_salary[2], col = "red")
abline(the_coefs_salary[1] + the_coefs_salary[4], the_coefs_salary[2], col = "black")
abline(the_coefs_salary[1] + the_coefs_salary[5], the_coefs_salary[2], col = "green")
legend(5, 230000, legend=c("Full", "Associate", "Assistant",
                       "Lecturer"),
       col=c("blue", "red", "black", "green"),
       title = "Rank of Professor", lty=1:2, cex=0.8)
```



Answer

It seems that using the same slope for a linear regression for Full Professors is poorly fitting the data. This is probably because Full Professors make much higher salaries than other professors and this is likely to increase much more with the amount of money a school has.

Part 2.5 (7 points): Fitting a slightly more complex model

Now fit a linear regression model for total salary as a function of log endowment size, but use separate y-intercepts **and slopes** for each of the 4 faculty ranks. Then answer the following questions:

1) How much of the total sum of squares of faculty salary is this model capturing?
2) Based on this model, if an Associate professor and Full professor both worked at a University that had an endowment of a million dollars, who would get paid more and by how much? Does this seem realistic?

```
(interaction_fit <- lm(salary_tot ~ log_endowment*rank_name, data = IPED_3))
```

```
##
## Call:
## lm(formula = salary_tot ~ log_endowment * rank_name, data = IPED_3)
##
## Coefficients:
##                        (Intercept)                        log_endowment
##                          -231986                             40888
##               rank_nameAssociate                rank_nameAssistant
##                           125551                            146880
##                rank_nameLecturer  log_endowment:rank_nameAssociate
##                           200710                            -18369
## log_endowment:rank_nameAssistant   log_endowment:rank_nameLecturer
##                           -22482                            -30100
```

```
summary(interaction_fit)
```

```
## 
## Call:
## lm(formula = salary_tot ~ log_endowment * rank_name, data = IPED_3)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -46914  -9581  -2180   6387  99678 
## 
## Coefficients:
##                                  Estimate Std. Error t value
## (Intercept)                       -231986       9778 -23.726
## log_endowment                       40888       1165  35.099
## rank_nameAssociate                 125551      13987   8.976
## rank_nameAssistant                 146881      14124  10.400
## rank_nameLecturer                  200710      20166   9.953
## log_endowment:rank_nameAssociate   -18369       1665 -11.033
## log_endowment:rank_nameAssistant   -22482       1681 -13.377
## log_endowment:rank_nameLecturer    -30100       2311 -13.025
##                                             Pr(>|t|)    
## (Intercept)                      <0.0000000000000002 ***
## log_endowment                    <0.0000000000000002 ***
## rank_nameAssociate               <0.0000000000000002 ***
## rank_nameAssistant               <0.0000000000000002 ***
## rank_nameLecturer                <0.0000000000000002 ***
## log_endowment:rank_nameAssociate <0.0000000000000002 ***
## log_endowment:rank_nameAssistant <0.0000000000000002 ***
## log_endowment:rank_nameLecturer  <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 15910 on 800 degrees of freedom
## Multiple R-squared:  0.7795, Adjusted R-squared:  0.7776 
## F-statistic:   404 on 7 and 800 DF,  p-value: < 0.00000000000000022
```

```
(the_coefs_interaction <- coef(interaction_fit))
```

```
##                      (Intercept)                    log_endowment 
##                       -231985.86                         40888.26 
##               rank_nameAssociate               rank_nameAssistant 
##                        125550.89                        146880.48 
##                rank_nameLecturer log_endowment:rank_nameAssociate 
##                        200710.11                        -18368.58 
## log_endowment:rank_nameAssistant  log_endowment:rank_nameLecturer 
##                        -22481.85                        -30100.41
```

```
predict(interaction_fit, (data.frame(log_endowment = 6, rank_name = "Full")))
```

```
##        1
## 13343.68
```

```
predict(interaction_fit, (data.frame(log_endowment = 6, rank_name = "Associate")))
```

```
##        1
## 28683.11
```

Answers

1) The adjusted R-squared value is 0.7776, so the model is capturing 77.76% of the sum of squares.
2) The Full professor would get paid $13343.68 and the Associate Professor would get paid $28683. The Associate professor is getting paid $15339.43 more. This seems very unrealistic because a full professor should be getting paid more; also, both of these salaries are far below what any professor in the U.S. should be making.

Part 2.6 (6 points): Visualizing the model

Now again recreate the scatter plot you created in part 2.2 using the same colors. Now, however, also add on the regression line with different y-intercepts and slopes based on the model you fit in part 2.5 (again use the appropriate colors).

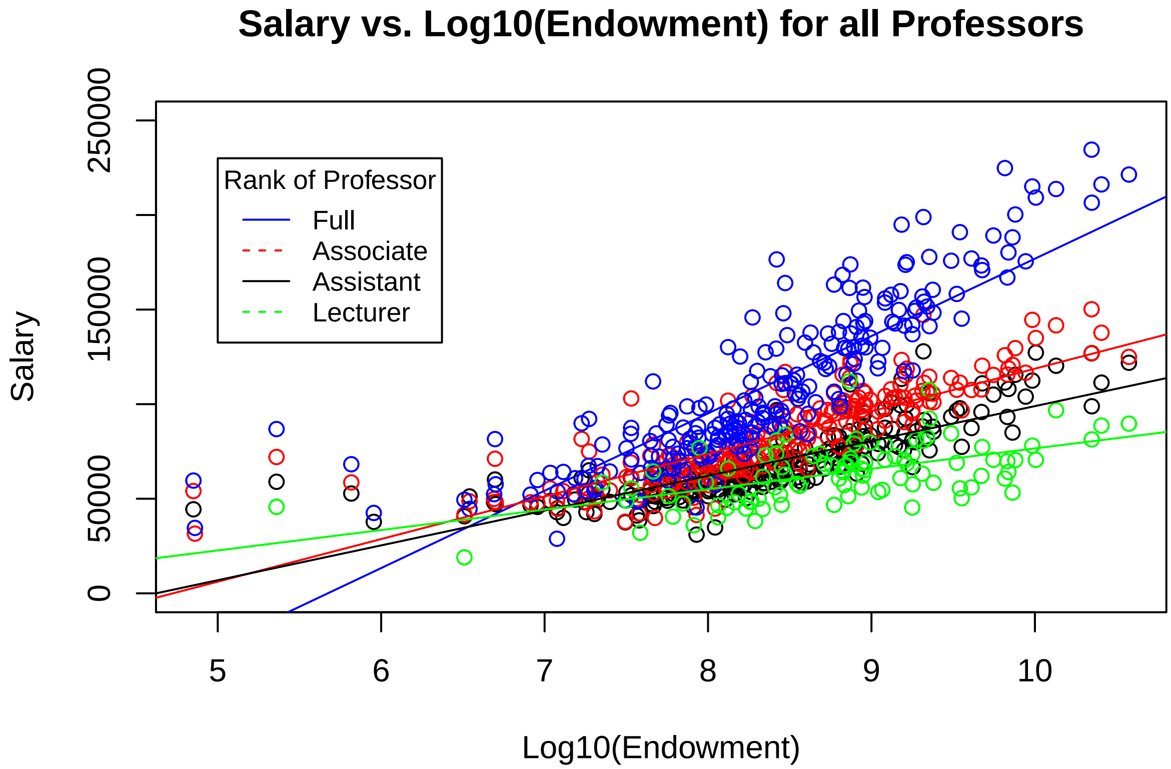
Does there seem to be an ordered relationship between ranks and how faculty salary increases with endowment?

```
plot(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Assistant"),
     ylim = c(0, 250000), ylab = "Salary", xlab = "Log10(Endowment)",
     main = "Salary vs. Log10(Endowment) for all Professors")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Associate"), col = "red")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Full"), col = "blue")
points(salary_tot ~ log_endowment, data = filter(IPED_3, rank_name == "Lecturer"), col = "green")

the_coefs_interaction
```

```
##                      (Intercept)                    log_endowment
##                       -231985.86                         40888.26
##               rank_nameAssociate               rank_nameAssistant
##                        125550.89                        146880.48
##                rank_nameLecturer log_endowment:rank_nameAssociate
##                        200710.11                        -18368.58
## log_endowment:rank_nameAssistant  log_endowment:rank_nameLecturer
##                        -22481.85                        -30100.41
```

```
abline(the_coefs_interaction[1], the_coefs_interaction[2], col = "blue")
abline(the_coefs_interaction[1] + the_coefs_interaction[3],
       the_coefs_interaction[2] + the_coefs_interaction[6], col = "red")
abline(the_coefs_interaction[1] + the_coefs_interaction[4],
       the_coefs_interaction[2] + the_coefs_interaction[7], col = "black")
abline(the_coefs_interaction[1] + the_coefs_interaction[5],
       the_coefs_interaction[2] + the_coefs_interaction[8], col = "green")
legend(5, 230000, legend=c("Full", "Associate", "Assistant",
                           "Lecturer"),
       col=c("blue", "red", "black", "green"),
       title = "Rank of Professor", lty=1:2, cex=0.8)
```



Answer

There does seem to be an ordered relationship between the rank and the increase in salary. When the endowment is over about 10 million dollars, the Full Professor salary increases the fastest, followed by Associate, Assistant, and then Lecturer. This follows the ranking of actual Professors - the higher the rank, the faster salary will increase.

Part 2.7 (3 points): Comparing models

The model you fit in Part 2.5 is nested within the model you fit in Part 2.3. Use an ANOVA to compare these models. Does adding the additional slopes for each rank seem to improve the model fit?

```
anova(intercept_fit, interaction_fit)
```

```
## Analysis of Variance Table
##
## Model 1: salary_tot ~ log_endowment + rank_name
## Model 2: salary_tot ~ log_endowment * rank_name
##   Res.Df          RSS Df   Sum of Sq      F                 Pr(>F)
## 1    803 270762445978
## 2    800 202593228230  3 68169217748 89.729 < 0.00000000000000022 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Answer Since the Residual Sum of Squares decreased with the second model, it seems that addition of additional slopes for each rank did improve the model fit. We also see that the p-value for the F-statistic is very small, showing there is a difference in the models' ability to explain the variance.

Part 2.8 (5 points): Improving the model

Can you think of any other ways to improve the model fit? Do an additional analysis where you adjust something about the model or the data to create a better model. Describe below what you did below and why.

```
poly_list <- c()
for (i in 1:5){
poly_list[[i]] <- lm(salary_tot ~ poly(log_endowment, degree = i)*rank_name, data = IPED_3)
}

(poly_fit_5 <- poly_list[[5]])
```

```
##
## Call:
## lm(formula = salary_tot ~ poly(log_endowment, degree = i) * rank_name,
##     data = IPED_3)
##
## Coefficients:
##                                          (Intercept)
##                                               112307
##                     poly(log_endowment, degree = i)1
##                                              1053277
##                     poly(log_endowment, degree = i)2
##                                               374555
##                     poly(log_endowment, degree = i)3
##                                               -69858
##                     poly(log_endowment, degree = i)4
##                                               -94644
##                     poly(log_endowment, degree = i)5
##                                                38416
##                                   rank_nameAssociate
##                                               -29025
##                                   rank_nameAssistant
##                                               -42309
##                                   rank_nameLecturer
##                                              -53699
## poly(log_endowment, degree = i)1:rank_nameAssociate
##                                             -486877
## poly(log_endowment, degree = i)2:rank_nameAssociate
##                                             -194300
## poly(log_endowment, degree = i)3:rank_nameAssociate
##                                                3122
## poly(log_endowment, degree = i)4:rank_nameAssociate
##                                               24440
## poly(log_endowment, degree = i)5:rank_nameAssociate
##                                                5863
## poly(log_endowment, degree = i)1:rank_nameAssistant
##                                             -596970
## poly(log_endowment, degree = i)2:rank_nameAssistant
##                                             -197550
## poly(log_endowment, degree = i)3:rank_nameAssistant
##                                               40694
## poly(log_endowment, degree = i)4:rank_nameAssistant
##                                               43646
## poly(log_endowment, degree = i)5:rank_nameAssistant
##                                              -29350
##  poly(log_endowment, degree = i)1:rank_nameLecturer
##                                             -761827
##  poly(log_endowment, degree = i)2:rank_nameLecturer
##                                             -325137
##  poly(log_endowment, degree = i)3:rank_nameLecturer
##                                              -28598
##  poly(log_endowment, degree = i)4:rank_nameLecturer
##                                              150916
##  poly(log_endowment, degree = i)5:rank_nameLecturer
##                                               10677
```

```
(summary(poly_fit_5))$adj.r.squared
```

```
## [1] 0.8547149
```

Answer

We can use a polynomial regression to try and improve the model fit. We can see from the scatterplot above that there is some curvature to the data, so I thought fitting a polynomial regression might help to address this problem. When I fitted a polynomial regression with degree 5, the R-squared value was 0.8547, so it seems to fit the data better than the linear.

Part 2.9 (5 points): Further explorations

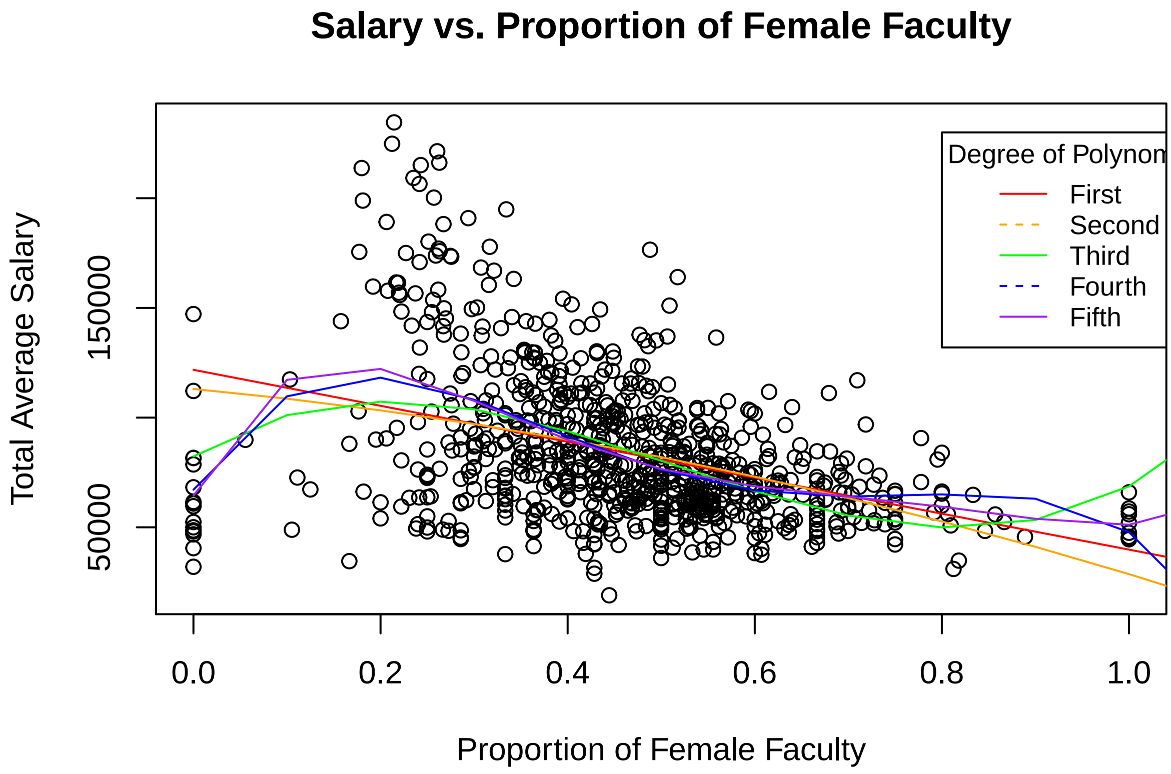
Do an additional exploration or model of the data and report something else interesting.

```
IPED_4 <- IPED_3 %>% mutate(prop_women = num_faculty_women/num_faculty_tot,
                           prop_men = num_faculty_men/num_faculty_tot)
plot(salary_tot ~ prop_women, data = filter(IPED_4),
     ylab = "Total Average Salary", xlab = "Proportion of Female Faculty",
     main = "Salary vs. Proportion of Female Faculty")
legend(0.8, 230000, legend=c("First", "Second", "Third",
                       "Fourth", "Fifth"),
       col=c("red", "orange", "green", "blue", "purple"),
       title = "Degree of Polynomial", lty=1:2, cex=0.8)

poly_models_4 <- list()
for (i in 1:5){
  poly_models_4[[i]] <- lm(salary_tot ~ poly(prop_women, degree = i), data = IPED_4)
}

predict_df_1 <- data.frame(prop_women = seq(0, 13, by = .1))
for(i in 1:5){

  y_vals_predicted <- predict(poly_models_4[[i]], newdata = predict_df_1)
  points(predict_df_1$prop_women, y_vals_predicted,
     type = "l", col = the_cols[i])
}
```



```
r_squared <- c()
r_adj <- c()
for (i in 1:5){
  r_squared[i] <- summary(poly_models_4[[i]])$r.squared
  r_adj[i] <- summary(poly_models_4[[i]])$adj.r.squared
}

r_squared
```

```
## [1] 0.1609012 0.1649185 0.2193693 0.2515759 0.2546323
```

```
r_adj
```

```
## [1] 0.1598601 0.1628438 0.2164565 0.2478478 0.2499853
```

```
summary(poly_models_4[[1]])
```

```
## 
## Call:
## lm(formula = salary_tot ~ poly(prop_women, degree = i), data = IPED_4)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -89771 -18990  -4469  15511 130351 
## 
## Coefficients:
##                               Estimate Std. Error t value
## (Intercept)                      84037       1088   77.24
## poly(prop_women, degree = i)  -384495      30928  -12.43
##                                          Pr(>|t|)    
## (Intercept)                  <0.0000000000000002 ***
## poly(prop_women, degree = i) <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 30930 on 806 degrees of freedom
## Multiple R-squared:  0.1609, Adjusted R-squared:  0.1599 
## F-statistic: 154.6 on 1 and 806 DF,  p-value: < 0.00000000000000022
```

Answer I compared the proportion of female faculty members at each university (using IPED_3 which already filtered out the four levels of professors, and the types of colleges). Then I fitted polynomial regressions up to degree five. For the linear regression I found a negative slope of -364495. This seemed to be statistically significant as the p-value was very close to 0, showing that there is some negative relationship between the proportion of female faculty memebers and the average salary at the school. However, none of the polynomial models seemed to fit the data well as all R-squared values were less than 0.3. Looking at the data points I believe there is some correlation though, so I think this may be worth looking more into. Perhaps there really is a wage gap, or perhaps the colleges with more female faculty are all smaller colleges with less money to pay professors.

Part 3: More data wrangling

Thanksgiving is coming up which means a lot of Americans will be traveling. In particular, since New Haven is relatively close to New York City, so it is likely that a number of people will be flying out of airports in the New York City area for the holiday. A major frustration to flying is when a flight is delayed. Let's use dplyr to do some quick explorations of the data to some ways to potentially avoid flight delays.

Let's start by loading data for flights leaving New York City in 2013. Use *? flights* for more information about the data set. You don't need to modify anything on the code below.

```
#install.packages("nycflights13")
# get the flight delays data and load dplyr
require("nycflights13")
data(flights)
data(airlines)   # the names of the airline carriers
```

Part 3.1 (5 points): Flights that start off with a delay might end up making up some time during the course of the flight. Test whether this is true on average. Hint: only use flights that have positive departure delay.

```
delayed_flights <- flights %>% filter(dep_delay > 0)

mean(delayed_flights$dep_delay - delayed_flights$arr_delay, na.rm = TRUE)
```

```
## [1] 4.607264
```

```
median(delayed_flights$dep_delay - delayed_flights$arr_delay, na.rm = TRUE)
```

```
## [1] 7
```

```
dep_delayed <- delayed_flights$dep_delay
arr_delayed <- delayed_flights$arr_delay

t.test(dep_delayed, arr_delayed)
```

```
##
##  Welch Two Sample t-test
##
## data:  dep_delayed and arr_delayed
## t = 21.291, df = 254640, p-value < 0.00000000000000022
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  4.306395 5.179653
## sample estimates:
## mean of x mean of y
##  39.37323  34.63021
```

Answers:

The mean difference between departure delays and arrival delays is 4.607, meaning on average, a delayed flight makes up 4.607 minutes during the course of a flight. The median of this value is 7. The t-test shows there is a statistically significant difference between the means so it does seem to be true on average.

Part 3.2 (5 points): One way to avoid being delayed would be to avoid the worst airlines. Which airline had the longest arrival delays on average, and how long was this average delay? Use the *airlines* data frame to figure out which airline each carrier code corresponds to.

```
# get the average delay for each airline

avg_arr_delay <- flights  %>%  group_by(carrier) %>%
   summarize(mean_arr_delay = mean(arr_delay, na.rm = TRUE))
arrange(avg_arr_delay, desc(mean_arr_delay))[1,]
```

```
## # A tibble: 1 x 2
##   carrier mean_arr_delay
##   <chr>            <dbl>
## 1 F9                21.9
```

```
carrier <- toString(avg_arr_delay[which.max(avg_arr_delay$mean_arr_delay),][1])
(airlines[which(airlines$carrier == carrier),])[2]
```

```
## # A tibble: 1 x 1
##   name
##   <chr>
## 1 Frontier Airlines Inc.
```

Answers:
Frontier Airlines had the longest average delay of 21.92 minutes.

Part 3.3 (5 points): Another way to avoid flight delays would be to avoid particularly bad times to fly. Which month of the year had the longest departure delays? Also report which hour of the day had the longest departure delays. Finally, report how many flights left at the hour of the day that had the longest delay and what the average deley was at that time.

```
(month_delay <- flights  %>%  group_by(month) %>%
   summarize(mean_arr_delay = mean(arr_delay, na.rm = TRUE))
 %>% arrange(desc(mean_arr_delay)))[1,]
```

```
## # A tibble: 1 x 2
##   month mean_arr_delay
##   <int>          <dbl>
## 1     7           16.7
```

```
(hour_delay <- flights  %>%  group_by(hour) %>%
   summarize(mean_arr_delay = mean(arr_delay, na.rm = TRUE))
  %>% arrange(desc(mean_arr_delay)))[1,]
```

```
## # A tibble: 1 x 2
##    hour mean_arr_delay
##   <dbl>          <dbl>
## 1    21           18.4
```

```r
dim(filter(flights, hour == 21))
```

```
## [1] 10933    19
```

Answers:

July had the longest departure delay of 16.71 minutes.

The hour with the longest departure delays was 21:00 (9pm) with a time of 18.39 minutes. There were 10933 flights that left at this time.

Reflection (3 points)

Please reflect on how the homework went by going to Canvas, going to the Quizzes link, and clicking on Reflection on homework 9